

Definition of Done

Nothing we build is called done because it works on a laptop. It is done when every applicable item below is checked, the evidence is linked, and you have signed it off. This is the standard we hold ourselves to on every engagement.

At a glance

The checklist applied before any sprint, feature or release is called done.

- Seven sections: scope, code, security, AI-specific, performance and accessibility, operations, handover
- Every item is a yes or no; "mostly" is a no
- Signed off per release by you and by me

Read in 10 minutes. Use section 04 only when the work includes models or agents.

How to use this document

- We agree this checklist with you at the start of the engagement and record any changes in the fields below. Items can be added or tightened per engagement. They are not removed silently.
- It applies at three levels: each feature or ticket, each sprint, and each release. Section 04 applies only when the work includes models, prompts or agents. Section 05 applies to anything with a user interface or a public endpoint.
- Each item is a yes/no question. "Mostly" is a no. If an item does not apply, we write *N/A* and one line saying why.
- Evidence lives next to the work: pull request links, test output, dashboards, documents. The hint under an item says what we usually attach.
- The sign-off at the end is per release. Sprint and feature sign-offs are recorded in the project tracker.

Engagement

Repository / tracker

Agreed test level (unit / integration /
end-to-end)

Agreed environments (dev / staging /
production)

Changes to this checklist for this
engagement

01 **Scope and acceptance**

- ☐ Acceptance criteria are written in plain language, attached to the ticket, and were agreed with you before work started.

Each
criterion
can
be
answered
yes
or
no
by
someone
who
did
not
build
the
feature.

-
- ☐ Every acceptance criterion has been demonstrated, and the demonstration is recorded or was witnessed by your named reviewer.
-
- ☐ Anything out of scope is listed on the ticket, so that "not done" and "not asked for" cannot be confused later, and any deviation from the original criteria was agreed in writing before it was built, not explained after.
-
- ☐ Open questions raised during the work are either resolved or logged as follow-up tickets with an owner.
-

02 Code and quality

- ☐ The change was reviewed by a second person, and every review comment is resolved or explicitly deferred with a reason.

Link
to
the
pull
request.
On
a
one-
person
engagement,
we
say
so
here
and
agree
a
substitute:
a
recorded
self-
review
or
a
review
by
your
engineer.

- ☐ Automated tests exist at the level agreed for this engagement and pass in CI, not only locally. They cover the failure paths, not only the happy path: invalid input, empty results, timeouts, and permission denied.
- ☐ Linting and type checks pass with zero errors, and no rule was disabled to make that true without a comment saying why.
- ☐ Dependencies were audited for known vulnerabilities, and any high or critical finding is fixed, upgraded, or accepted by you in writing.
- ☐ Dependency versions are pinned or locked, and the build is reproducible from a clean checkout using only the documented steps, so that a fresh install reproduces the tested build.
- ☐ Database or schema changes ship with a migration and a tested way to reverse it.
- ☐ Commit history is readable: each commit says what changed and why, and no commit is titled "fix" or "wip".

03 Security and privacy

- ☐ Every endpoint and screen was checked for two things separately: is the user who they claim to be (authentication), and are they allowed to do this (authorisation)? Authorisation was also tested negatively: a user of one account or role tried to reach another's data and was refused.

☐ All input from outside the system is validated at the boundary: type, length, format and allowed values.

This
includes
file
uploads,
webhooks,
query
strings
and
anything
a
model
produces.

☐ Personal data handled by the change is listed: what fields, where they are stored, how long they are kept, and who can see them. It is not written to logs, error reports, analytics or model prompts unless you have approved that use.

If
the
answer
is
"none",
say
so.
That
is
a
valid
and
useful
answer.

☐ Secrets, credentials, tokens and private keys are held in a vault or managed secret store, never in configuration files, environment files in the repo, chat messages, or the repository and its history.

A
secret-
scanning
tool
ran
on
the
full
history,
not
only
the
diff.

☐ Every service account, API key and role has the least privilege needed for its job, and unused permissions were removed.

☐ Security headers are set on web responses: content security policy, strict transport security, frame options, content type options, and referrer policy.

☐ Transport is encrypted end to end, and data at rest is encrypted where the platform allows it.

☐ Rate limiting or abuse protection exists on any endpoint that is public, costs money to call, or sends messages.

☐ Error messages shown to users do not reveal stack traces, internal hostnames, or query text.

04 AI-specific items

Applies when the work includes a model, a prompt, a retrieval pipeline or an agent. Skip and mark N/A otherwise.

☐ An evaluation set exists, is versioned with the code, and was agreed with you as representative of real use.

It
includes
cases
the
system
should
refuse
or
escalate,
not
only
cases
it
should
answer.

☐ The evaluation runs automatically and passes the threshold agreed for this engagement. The score is recorded with the release.

☐ Every prompt and system instruction is versioned in the repository, and the version in production can be identified from any single log entry.

☐ Guardrails are in place and tested: input filtering, output checks, and refusal behaviour for content outside the intended use.

☐ Any action with a side effect (sending a message, changing a record, spending money, calling an external system) passes through an approval gate or an explicit allow-list agreed with you.

The
default
is:
read
freely,
write
only
with
approval.
Widening
that
is
a
documented
decision,
not
a
convenience.

☐ Cost per request and per day was measured on the evaluation set and on a realistic load, and stays within the budget agreed for this engagement.

☐ Latency was measured at the median and the 95th percentile, and stays within the target agreed for this engagement.

- ☐ Fallback behaviour is defined and tested for: model timeout, provider outage, rate limit, malformed output, and low-confidence answers. The user sees something honest, not a spinner.
- ☐ Every model call is logged with: timestamp, prompt version, model identifier, input reference, output reference, tokens, cost, and the user or process that triggered it.

Logs
exclude
raw
personal
data
unless
you
approved
it
under
section
03.

- ☐ Actions taken by an agent are written to an audit log that a person can read to reconstruct what happened and why.
- ☐ The model, provider and data-retention terms in use are named in the documentation, and you have confirmed they are acceptable for your data.
- ☐ The system says what it is. Users are told when they are interacting with an automated system where that is not obvious.

05 Performance and accessibility

- ☐ Performance budgets were agreed for this engagement and are met on the target environment, not on a developer machine; for web work, Core Web Vitals (LCP, INP, CLS) are in the "good" range on a mid-range mobile device over a throttled connection.

Typical
budgets:
page
weight,
time
to
interactive,
API
response
at
the
95th
percentile.

- ☐ For any user interface, WCAG 2.1 AA checks were run: colour contrast, keyboard-only navigation, visible focus, form labels, image alt text, and correct heading order. The interface was also used with a screen reader for at least the primary journeys, and nothing was unreachable or unnamed.
- ☐ The interface was tested on real mobile devices at the smallest supported width, including with the on-screen keyboard open.
- ☐ Load or capacity was tested at the level agreed for this engagement, the system degrades gracefully beyond it, and no database query on a hot path performs a full scan or grows with the size of the result set without a limit.

06 Operations

- ☐ The change is deployed to the environment agreed for this release by the pipeline, not by hand, and the infrastructure is defined in code and can be recreated from the repository.
- ☐ Monitoring covers availability, error rate and latency, and each has an alert that reaches a named person. Alerts were tested by triggering one deliberately, and the right person received it.

An
alert
nobody
receives
is
a
log
line.

- ☐ Backups run on a schedule, and a restore was actually performed and verified at least once, not assumed.
- ☐ A rollback path exists, is documented, and was rehearsed. The time to roll back is known.
- ☐ A runbook exists for this system covering: how to deploy, how to roll back, what the alerts mean, common failures and their fixes, and who to call.
- ☐ Logs are centralised, searchable, and retained for the period agreed for this engagement.
- ☐ Recurring costs of the deployed system are documented, and you know what they are before go-live.

07 Handover

- ☐ Documentation is updated to match what was shipped: README, architecture notes, API reference, and the runbook.

A
document
that
describes
the
previous
version
is
worse
than
no
document.

- ☐ The repository lives in your organisation's account, with your people as owners and us as collaborators you can remove at any time, and all credentials, domains, cloud accounts and third-party services are registered to you, with any created under our name transferred.
- ☐ Our access is listed, and the plan to reduce or revoke it after the engagement is agreed.

- ☐ A demonstration of the delivered work was held with you or recorded for you, the recording or notes are linked here, and your named person can deploy, roll back and read the monitoring without us on the call.

We
test
this
by
watching
them
do
it,
not
by
asking.

- ☐ Known limitations and deferred items are written down with an owner and a next step, so nothing is discovered later as a surprise.
- ☐ You have signed off in writing below.

08 Sign-off

Signing confirms that every applicable item above is checked with evidence, and that items marked N/A carry a reason both parties accept.

Release / sprint / feature

Version or tag

Items marked N/A and why

Deferred items (ticket references)

Evidence location

Signed for Onebit-AI · name · date

Signed for the client · name · date

Defaults in this checklist (test level, budgets, thresholds, retention periods, environments) are agreed between Onebit-AI and the client per engagement and recorded in the fields at the top of this document.